King Fahd University of Petroleum and Minerals

Aerospace Engineering Department

Flight Dynamics and Control I

(AE 540)

Term Project

SALMAN AL-FIFI

ID # 991694

Objectives:

Based on (**DHC-2**) aircraft the following is required:

- Finding the differential equations of motion that governs this aircraft by writing a Matlab code and plotting the response.
- Calculating all flying coefficients and all flying derivatives to generate the A and B matrices for this aircraft. And finding trim values.
- Designing an altitude hold autopilot by using the real A and B matrices of this aircraft by classical control theory for Longitudinal.
- Designing an altitude hold autopilot by using the real A and B matrices of this aircraft by modern control theory Longitudinal.

Background information:

Aircraft has many aspects that should be maintained properly in design, according to the type of the aircraft; all flying qualities would be different from one airplane to another based on the specific mission of that airplane. For example, the general aviation aircraft must be stable in all conditions to be more secure and comfortable for passengers. On the other hand, the fighter aircraft needs to be sometimes unstable to be more available for maneuvering. These differences occur because of stability derivatives that differ from one aircraft to another.
In this project, we will consider this issue and we will calculate these derivatives in a specific speed and then monitoring the results and see whether they give reasonable solutions and expected results.
In my own project I will consider the fighter aircraft **(DHC-2)** to find what stated in the objectives.

Part 1: (finding the 12 differenatial equations by using Matlab)

The Matlab code is:

- The function file is:

```
%%  soving the 12 Differentail equations of aircraft system

%%THE FUNCTION M-FILE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function sadot=salman(t,sa);

global vdot

%DENSITY
rho=1.024;

%WING SPAN
b=14.63;

%MEAN AERODYNAMIC CHORD
cbar=1.5875;

%WING AREA
s=23.23;

%DERIVATIVES
del_r = 0;
del_f = 0;
del_a = 0;
del_e = 0;
dpt = -0.35;

%VELOCITY COMPONENT ALONG ROLL,PITCH AND YAW AXIS
u=sa(1); v=sa(2); w=sa(3);

%ANGULAR RATES ALONG ROLL,PITCH AND YAW AXIS
p = sa(4);  q = sa(5); r = sa(6);

%EULER ANGLES ALONG ROLL,PITCH AND YAW AXIS
phi = sa(7); theta = sa(8); psi = sa(9);
```

```
du = vdot(1); dv =vdot(2); dw = vdot(3);

%FREE SREAM VELOCITY
V = sqrt(u^2+v^2+w^2);

%RATE OF CHANGE OF FREE SREAM VELOCITY
dV = (u*du+v*dv+w*dw)/V;

%ANGLE OF ATTACK
alpha=atan(w/u);

%SIDESLIP ANGLE
beta=asin(v/V);

%RATE OF CHANGE OF SIDESLIP ANGLE
betadot = (V*cos(beta*dv)-v*cos(beta*dV))/(V*cos(beta))^2;

%DYNAMIC PRESSURE%
Q=0.5*rho*V^2;


%CALCULATIONS OF COEFFICIENTS
Cxa = -0.03554 + 0.002920*alpha + 5.459*alpha^2 - 5.162*alpha^3 -
0.6748*(q*cbar/V) + 0.03412*del_r - 0.09447*del_f + 1.106*alpha*del_f;

Cya = -0.002226 - 0.7678*beta - 0.1240*((p*b)/(2*V)) + 0.3666*((r*b)/(2*V)) -
0.02956*del_a + 0.1158*del_r + 0.5238*del_r*alpha - 0.1600*((betadot*b)/(2*V));

Cza = -0.05504 - 5.578*alpha + 3.442*alpha^3 - 2.988*(q*cbar/V) - 0.3980*del_e -
15.93*del_e*beta^2 - 1.377*del_f - 1.261*alpha*del_f;

Cla = 0.0005910 - 0.06180*beta - 0.5045*((p*b)/(2*V)) + 0.1695*((r*b)/(2*V)) -
0.09917*del_a + 0.006934*del_r - 0.08269*del_a*alpha;

Cma = 0.09448 - 0.6028*alpha - 2.140*alpha^2 - 15.56*(q*cbar/V) - 1.921*del_e +
0.6921*beta^2 - 0.3118*((r*b)/(2*V)) + 0.4072*del_f;

Cna = -0.003117 + 0.006719*beta - 0.1585*((p*b)/(2*V)) - 0.1112*((r*b)/(2*V)) -
0.003872*del_a - 0.08265*del_r + 0.1595*(q*cbar/V) + 0.1373*beta^3;

Cxp = 0.1161*dpt + 0.1453*alpha*dpt^2;

Cyp = 0;

Czp = -0.1563*dpt;

Clp = -0.01406*alpha^2*dpt;

Cmp = -0.07895*dpt;
```

```
Cnp = -0.003026*dpt^3;


%FORCE EQUATIONS

%AXIAL FORCE
fx = (Cxa+Cxp)*Q*s;

%SIDE FORCE
fy = Cya*Q*s;

%NORMAL FORCE
fz = (Cza+Czp)*Q*s;

%ROLLING MOMENT
L = (Cla+Clp)*Q*s*b;

%PITCHING MOMENT
M = (Cma+Cmp)*Q*s*cbar;

%YAWING MOMENT
N = (Cna+Cnp)*Q*s*b;

%MOMENT OF INERTIA
ix=5368.39;iy=6928.93;iz=11158.75;

%PRODUCT OF INERTIA
ixz=117.64;ixy=0;iyz=0;

%ACCELERATION DUE TO GRAVITY
g=9.81;

%MASS OF THE AIRCRAFT
m=2288.231;


%VELOCITY EQUATIONS
du= -q*w+r*v-g*sin(theta)+(fx/m); %d(u)
dv= (-r*u)+(p*w)+(fy/m)+(g*cos(theta)*sin(phi)); %d(v)
dw= (-p*v)+(q*u)+(fz/m)+(g*cos(theta)*sin(phi)); %d(w)


%MOMENT EQUATIONS
dp= ((q*r*iz)*(iy-iz)+(p*q*ixz)*(ix-iy)+(p*q*ixz*iz)-
(q*r*ixz*ixz)+(L*iz)+(N*ixz))/(ix*iz-ixz^2);
dq= (M/iy)-(((q*r)/iy)*(ix-iz))-(ixz/iy)*(p^2-r^2);
dr= ((q*r*ixz)*(iy-iz)+(p*q*ix)*(ix-iy)+(p*q*ixz*ixz)-
(q*r*ix*ixz)+(L*ixz)+(N*ix))/(ix*iz-ixz^2);
```

```
%KINETIC EQUATIONS
dphi= p+q*sin(phi)*tan(theta);
dtheta= q*cos(phi)-r*sin(phi); %d(theta)
dpsi= (q*sin(phi)*sec(theta))+(r*cos(phi)*sec(theta));


%FIXED FRAME AIRCRAFT VELOCITY IN TERMS OF EULER ANGLES AND
BODY VELOCITY COMPONENT
dx= (u*cos(theta)*cos(psi)) + (v*sin(phi)*sin(theta)*cos(psi)) - (v*cos(phi)*sin(psi))
+ (w*cos(psi)*sin(theta)*cos(psi)) + (w*sin(phi)*sin(psi));
dy= (u*cos(theta)*sin(psi)) + (v*sin(phi)*sin(theta)*sin(psi)) + (v*cos(phi)*cos(psi))
+ (w*cos(phi)*sin(theta)*sin(psi)) - (w*sin(phi)*cos(psi));
dz= (-u*sin(theta)) + (v*sin(phi)*cos(theta)) + (w*cos(phi)*cos(theta));



sadot=[du dv dw dp dq dr dphi dtheta dpsi dx dy dz]';

vdot(1) = du; vdot(2) = dv; vdot(3)= dw;
```

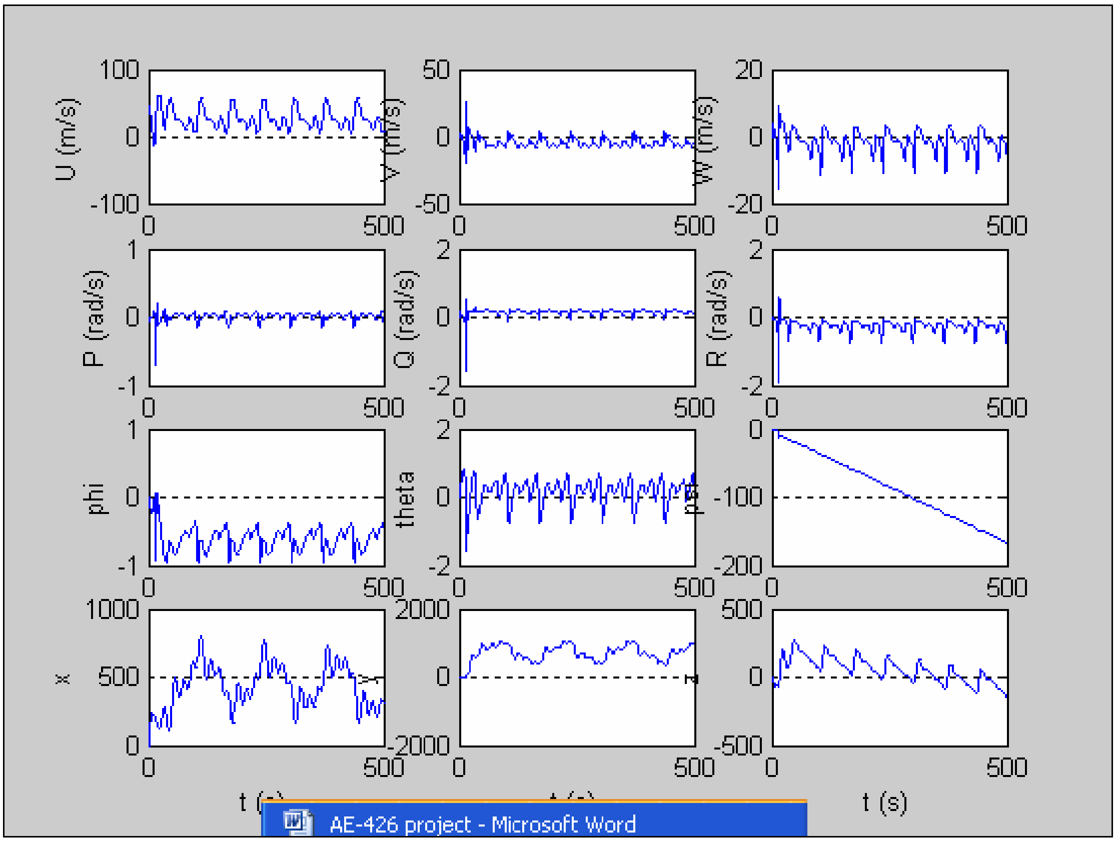
➔ The call function file is:

```
%% SIMULATION OF KINEMATIC AND DYNAMIC EQUATION OF BEAVER
MODEL AIRCRAFT
%% CALLING FUNCION PROGRAM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all; clear all; clc;
global vdot
vdot=[0 0 0];
tspan=[0 500]; % time interval

ro=[45 0 0 0 0 0 0 0 0 0 0 0]; % initial conditions

[t,sa]=ode45('A',tspan,ro);


subplot(4,3,1),plot(t,sa(:,1));xlabel('t (s)'); ylabel('U (m/s)');grid on;
subplot(4,3,2),plot(t,sa(:,2));xlabel('t (s)'); ylabel('V (m/s)');grid on;
subplot(4,3,3),plot(t,sa(:,3));xlabel('t (s)'); ylabel('W (m/s)');grid on;

subplot(4,3,4),plot(t,sa(:,4));xlabel('t (s)'); ylabel('P (rad/s)');grid on;
subplot(4,3,5),plot(t,sa(:,5));xlabel('t (s)'); ylabel('Q (rad/s)');grid on;
subplot(4,3,6),plot(t,sa(:,6));xlabel('t (s)'); ylabel('R (rad/s)');grid on;

subplot(4,3,7),plot(t,sa(:,7));xlabel('t (s)'); ylabel('phi');grid on;
subplot(4,3,8),plot(t,sa(:,8));xlabel('t (s)'); ylabel('theta');grid on;
```

```
subplot(4,3,9),plot(t,sa(:,9));xlabel('t (s)'); ylabel('psi');grid on;

subplot(4,3,10),plot(t,sa(:,10));xlabel('t (s)'); ylabel('x');grid on;
subplot(4,3,11),plot(t,sa(:,11));xlabel('t (s)'); ylabel('y');grid on;
subplot(4,3,12),plot(t,sa(:,12));xlabel('t (s)'); ylabel('z');grid on;
```

❖ The responses:

The call function code is:



Part 2: (calculating the A & B-matrices and trim values of this system by Matlab)

- The Matlab code is:

  - Longitudinal A & B matrices code is:

```
%%%%%%% LONGITUDINAL MATRIX%%%%%%%


clear all
clc

% Gravity [m / sq meter]
   g=9.81;

% Air density [kg / cubic meter]
    rho = 1.024;
% Flight Path Angle [deg]
         u0 = 45;
    M = 0.135;
    Q = 0.5*rho*u0^2;

% Aircraft mass [kg]
    m = 2288.231;
% Aircraft moments of inertia [kg-sq.m.]
    Ix = 5368.39;
    Iy = 6928.93;
    Iz = 11158.75;
% Aircraft product of inertia [slugs-sq.ft.]
         Izx = 117.64;
% Aircraft wing reference area [sq.m.]
         S = 23.23;
         QS = Q*S;
% Aircraft mean wing chord [m]
         cbar=1.5875;
% Aircraft wing span [m]
         b = 14.63;

                  CLo = 0.4;
                  CDo = 0.03554;
                  CLalpha = 5.578;
                  CDalpha = 0.4426;
                  CMalpha = -0.789;
                  CMalphadot = -170;
                  CMq = -441.07;
                  CLm = 0.0;
                  CDm = 0.0;
                  CMm = 0;
```

```
            CTu = -CDo;
            CXdeltar = 0.03412;
            CZdeltae = -0.3980;
            CMdeltae = -1.924;
            CXdeltaT = 0.1161;
            CZdeltaT = -0.1563;
            CMdeltaT = -0.07895;

            Xu = (-(CDm*M +2*CDo)+CTu)*QS/(m*u0);
            Xw = -(CDalpha-CLo)*QS/(m*u0);
            Xdelta = CXdeltar*QS/m;
    XdeltaT = CXdeltaT*QS/m;

            Zu = -(CLm*M+2*CLo)*QS/(m*u0);
            Zw = -(CLalpha+CDo)*QS/(m*u0);
    Zalpha = u0*Zw;

            Zdelta = CZdeltae*QS/m;
    ZdeltaT = CZdeltaT*QS/m;

            Mu = (CMm*M)*(QS*cbar/(u0*Iy));
            Mw = CMalpha*(QS*cbar/(u0*Iy));
    Malpha = u0*Mw;
            Mwdot = CMalphadot*(cbar/(2*u0))*(QS*cbar/(u0*Iy));
    Malphadot = u0*Mwdot;
            Mq = CMq*cbar/(2*u0)*(QS*cbar/Iy);
            Mdelta = CMdeltae*(QS*cbar/Iy);
    MdeltaT = CMdeltaT*(QS*cbar/Iy);

A_long=     [Xu Xw 0 -g
            Zu Zw u0 0
            Mu+Mwdot*Zu Mw+Mwdot*Zw Mq+Mwdot*u0 0
            0 0 1 0]
B_long = [Xdelta XdeltaT
    Zdelta ZdeltaT
    Mdelta+Mwdot*Zdelta MdeltaT+Mwdot*ZdeltaT
    0 0]
[eigenvectors, eigenvalues] = eig(A_long);
damp(eigenvalues),
```

- Longitudinal A & B matrices values and their eigenvalues and the damping ratios and the natural frequencies:

```
A_long =

  -0.0249   -0.0100         0          -9.8100
  -0.1871   -1.3130   45.0000               0
   0.0688    0.3860  -59.4777               0
        0         0    1.0000               0


B_long =

   0.3591    1.2220
  -4.1892   -1.6451
  -9.0765    0.1693
        0         0


      Eigenvalue                 Damping        Freq. (rad/s)

 -1.60e-002 + 5.19e-002i       2.95e-001       5.43e-002
 -1.60e-002 - 5.19e-002i       2.95e-001       5.43e-002
 -1.01e+000                    1.00e+000       1.01e+000
 -5.98e+001                    1.00e+000       5.98e+001
```

☒ Lateral A & B matrices code is:

```
%%%%%%% LATERAL MATRIX%%%%%%%

clear all
clc

% Gravity [m / sq meter]
   g=9.81;

% Air density [kg / cubic meter]
    rho = 1.024;
% Flight Path Angle [deg]
                                 u0 = 45;
    M = 0.135;
    Q = 0.5*rho*u0^2;

% Aircraft mass [kg]
    m = 2288.231;





% Aircraft wing reference area [sq.m.]
                                  S = 23.23;
                                  QS = Q*S;
% Aircraft mean wing chord [m]
                                  cbar=1.5875;
% Aircraft wing span [m]
                                  b = 14.63;





% Moment of inertia of the Aircraft
   Ix = 5368.39;                 Iy = 6928.93;      Iz = 11158.75;
   Ixy = 0;                         Ixz = 117.64;            Iyz = 0;

% Control Values
   dr = 0.0117;  df = 0;   da = 0.0109;   de = -0.0074;   dpt = -1.1092;





% Moment of inertia of the Aircraft
   Ix = 5368.39;                 Iy = 6928.93;      Iz = 11158.75;
   Ixy = 0;                         Ixz = 117.64;            Iyz = 0;
```

```
% Aerodynamics Forces Coefficients
  CX0   = -0.03554;    CZ0    = -0.05504;    Cm0  =  0.09448;
  CXa   =  0.002920;   CZa    = -5.578;      Cma  = -0.6028;
  CXa2  =  5.459;      CZa3   =  3.442;      Cma2 = -2.140;
  CXa3  = -5.162;      CZq    = -2.988;      Cmq  = -15.56;
  CXq   = -0.6748;     CZde   = -0.3980;     Cmde = -1.921;
  CXdr  =  0.03412;    CZdeb2 = -15.93;      Cmb2 =  0.6921;
  CXdf  = -0.09447;    CZdf   = -1.377;      Cmr  = -0.3118;
  CXadf =  1.106;      CZadf  = -1.261;      Cmdf =  0.4072;

  CY0   = -0.002226;   Cl0    =  0.0005910;  Cn0  = -0.003117;
  CYb   = -0.7678;     Clb    = -0.06180;    Cnb  =  0.006719;
  CYp   = -0.1240;     Clp    = -0.5045;     Cnp  = -0.1585;
  CYr   =  0.3666;     Clr    =  0.1695;     Cnr  = -0.1112;
  CYda  = -0.02956;    Clda   = -0.09917;    Cnda = -0.003872;
  CYdr  =  0.1158;     Cldr   =  0.006934;   Cndr = -0.08265;
  CYdra =  0.5238;     Cldaa  = -0.08269;    Cnq  =  0.1595;
  CYbdot= -0.1600;                           Cnb3 =  0.1373;

  CXdpt   =  0.1161;
  CXadpt2 =  0.1453;
  CZdpt   = -0.1563;
  Cla2dpt = -0.01406;
  Cmdpt   = -0.07895;
  Cndpt3  = -0.003026;




Ybeta=(Q*S*CYb)/m;


Yp=(Q*S*b*CYp)/(2*m*u0);

Lp=(Q*S*b^2*Clp)/(2*Ix*u0);

CYr=-2*(44/b)*CYb;

Yr=(Q*S*b*CYr)/(2*m*u0);

Lr=(Q*S*b^2*Clr)/(2*Ix*u0);

Yalphaa=(Q*S*CYda)/m;

Ydeltar=(Q*S*CYdr)/m;

Ndeltaa=(Q*S*b*Cnda)/Iz;
```

```
Nbeta=(Q*S*b*Cnb)/Iz;

Lbeta=(Q*S*b*Clb)/Ix;

Np=(Q*S*b^2*Cnp)/(2*Iz*u0);

Nr=(Q*S*b^2*Cnr)/(2*Iz*u0);

Ydeltar=(Q*S*CYdr)/m;

Ndeltar=(Q*S*b*Cndr)/Iz;

Ldeltaa=(Q*S*b*Clda)/Ix;

Ldeltar=(Q*S*b*Cldr)/Ix;

theta0=0;




A_lateral=                [Ybeta/u0 Yp/u0 -(1-Yr/u0) g*cos(theta0)/u0
                           Lbeta Lp Lr 0
                           Nbeta Np Nr 0
                           0 1 0 0]
B_lateral = [0 Ydeltar/u0
        Ldeltaa Ldeltar
       Ndeltaa Ndeltar
       0 0]
[eigenvectors, eigenvalues] = eig(A_lateral);
damp(eigenvalues),
```

- Lateral A & B matrices values and their eigenvalues and the damping ratios and the natural frequencies:

A_lateral =

-0.1796	-0.0047	-0.8244	0.2180
-4.0563	-5.3828	1.8085	0
0.2122	-0.8136	-0.5708	0
0	1.0000	0	0

B_lateral =

0	0.0271
-6.5092	0.4551
-0.1223	-2.6099
0	0

Eigenvalue	Damping	Freq. (rad/s)
-8.59e-002	1.00e+000	8.59e-002
-4.13e-001 + 8.77e-001i	4.26e-001	9.69e-001
-4.13e-001 - 8.77e-001i	4.26e-001	9.69e-001
-5.22e+000	1.00e+000	5.22e+000

- Trim values:

The function file is:

```
function f=trim(x)

% Air Density
rho=1.024;
% Mass of the Aircraft
m=2288.231;
% Acceleration
g=9.81;
% Angle
psi = 0;
% Side Slip Angle & Side Slip Angle Rate
beta=0;
betadot = 0;
% Trim Condition
u0  = 45;
u   = u0;
v   = 0;
w   = 0;
p   = 0;
q   = 0;
r   = 0;
df  = 0;
% Aircraft Velocity
V=sqrt(u^2 + v^2 + w^2);
% Aircraft Parameter
b=14.63;      cbar=1.5875;      S=23.23;
% Dynamic Pressure
Q=0.5*rho*V^2;

% Aerodynamics Forces Coefficients
  CX0   = -0.03554;   CZ0    = -0.05504;   Cm0  =  0.09448;
  CXa   =  0.002920;  CZa    = -5.578;     Cma  = -0.6028;
  CXa2  =  5.459;     CZa3   =  3.442;     Cma2 = -2.140;
  CXa3  = -5.162;     CZq    = -2.988;     Cmq  = -15.56;
  CXq   = -0.6748;    CZde   = -0.3980;    Cmde = -1.921;
  CXdr  =  0.03412;   CZdeb2 = -15.93;     Cmb2 =  0.6921;
  CXdf  = -0.09447;   CZdf   = -1.377;     Cmr  = -0.3118;
  CXadf =  1.106;     CZadf  = -1.261;     Cmdf =  0.4072;

  CY0   = -0.002226;  Cl0    =  0.0005910; Cn0  = -0.003117;
  CYb   = -0.7678;    Clb    = -0.06180;   Cnb  =  0.006719;
  CYp   = -0.1240;    Clp    = -0.5045;    Cnp  = -0.1585;
  CYr   =  0.3666;    Clr    =  0.1695;    Cnr  = -0.1112;
  CYda  = -0.02956;   Clda   = -0.09917;   Cnda = -0.003872;
  CYdr  =  0.1158;    Cldr   =  0.006934;  Cndr = -0.08265;
  CYdra =  0.5238;    Cldaa  = -0.08269;   Cnq  =  0.1595;
  CYbdot= -0.1600;                         Cnb3 =  0.1373;
```

```
 CXdpt   =  0.1161;
 CXadpt2 =  0.1453;
 CZdpt   = -0.1563;
 Cla2dpt = -0.01406;
 Cmdpt   = -0.07895;
 Cndpt3  = -0.003026;

% x(1) = alpha;
% x(2) = da;
% x(3) = dr;
% x(4) = de;
% x(5) = theta;
% x(6) = dpt;

% Aerodynamics Forces Coefficients
CXa = CX0 + CXa*x(1) + CXa2*x(1)^2 + CXa3*x(1)^3 + CXq*((q*cbar)/V) +
CXdr*x(3) + CXdf*df + CXadf*x(1)*df;
CYa = CY0 + CYb*beta + CYp*((p*b)/(2*V)) + CYr*((0.5*r*b)/V) + CYda*x(2) +
CYdr*x(3) + CYdra*x(3)*x(1) + CYbdot*((betadot*b)/(2*V));
CZa = CZ0 + CZa*x(1) + CZa3*x(1)^3 + CZq*((q*cbar)/V) + CZde*x(4) +
CZdeb2*x(4)*beta^2 + CZdf*df + CZadf*x(1)*df;
% Aerodynamics Moments Coefficients
Cla = Cl0 + Clb*beta + Clp*((p*b)/(2*V)) + Clr*((r*b)/(2*V)) + Clda*x(2) +
Cldr*x(3) + Cldaa*x(2)*x(1);
Cma = Cm0 + Cma*x(1) + Cma2*x(1)^2 + Cmq*((q*cbar)/V) + Cmde*x(4) +
Cmb2*beta^2 + Cmr*((r*b)/(2*V)) + Cmdf*df;
Cna = Cn0 + Cnb*beta + Cnp*((p*b)/(2*V)) + Cnr*((r*b)/(2*V)) + Cnda*x(2) +
Cndr*x(3) + Cnq*((q*cbar)/V) + Cnb3*beta^3;
% Propulsive Forces Coefficients
CXp = CXdpt*x(6) + CXadpt2*x(1)*x(6)^2;
CYp = 0;
CZp = CZdpt*x(6);
% Propulsive Moments Coefficients
Clp = Cla2dpt*x(1)^2*x(6);
Cmp = Cmdpt*x(6);
Cnp = Cndpt3*x(6)^3;

% Total forces
Fx=(CXa+CXp)*Q*S;
Fy=(CYa+CYp)*Q*S;
Fz=(CZa+CZp)*Q*S;
% Total moments
L=(Cla+Clp)*Q*S*b;
M=(Cma+Cmp)*Q*S*cbar;
N=(Cna+Cnp)*Q*S*b;

% Nonlinear System of Equations to be Solved
f(1) = Fx - m*g*sin(x(5));
f(2) = Fy + m*g*cos(x(5))*sin(psi);
f(3) = Fz + m*g*cos(x(5))*cos(psi);
```

```
f(4) = L;
f(5) = M;
f(6) = N;
```

The call function file is:

```
clear all
clc

x=[0 0 0 0 0 0];

x=fsolve('trim',x)
```

The trim values are:

```
% x(1) = alpha  = 0.1931;
% x(2) = da     = 0.0109;
% x(3) = dr     = 0.0117;
% x(4) = de     = -0.0074;
% x(5) = theta  = 0.0404;
% x(6) = dpt    = -1.1092;
```

Part 3: (Designing an altitude hold autopilot by using the real A and B matrices of this aircraft by classical control theory).

The A and B matrices are:

```
A=[-0.03893 5.4535 -0.40758 -9.8
   -0.008376 -1.2854 0.9764 -0.000001089
   0.013905 -6.7369 -3.0292 0
   0 0 1 0];

B=[-0.60796
   -0.092937
   -10.601
   0];
```

➔ To design an altitude hold autopilot, we only concentrate on the longitudinal mode and trying to control the pitching angle (θ) and the elevator deflection angle ($\alpha_{\delta E}$).

➔ The transfer function of the system is found by writing the following Matlab code:

```
A=[-0.03893 5.4535 -0.40758 -9.8
   -0.008376 -1.2854 0.9764 -0.000001089
   0.013905 -6.7369 -3.0292 0
   0 0 1 0];

B=[-0.60796
   -0.092937
   -10.601
   0];

u0=45;
C2_alpha=[0 1 0 0];
C1_q=[0 0 1 0];
D=0;

[num1,den1]=ss2tf(A,B,C1_q,D);

[num2,den2]=ss2tf(A,B,C2_alpha,D);




T1_q=SS2TF(A,B,C1_q,D);num1=[0  -10.6010  -13.4216   -1.0426
0.0000];
```

```
                den1=[1.0000    4.3535   10.6910    0.6385    0.7282];


T2_alpha=SS2TF(A,B,C2_alpha,D);num2ns=[0   -0.0929  -10.6309   -
0.4435   -0.8828];
                    den2ns=[1.0000    4.3535   10.6910    0.6385
0.7282];


 TF1=tf(num1,den1);
 TF2ns=tf(num2ns,den2ns);
 s=[1 0];
 num2=conv(s,num2ns);
 den2=den2ns;
 TF2=tf(num2,den2);
 TF3=TF1-TF2;
 num3=[0.0929 0.4343 -11.85 -56.28 -139.4 -9.973 -9.553 -0.1164 0];
 den3=[1 8.707 40.33 94.36 121.3 19.99 15.98 0.9299 0.5303];

 num4=[u0];den4=[1 0 0];
 TF4=tf(num4,den4);
num5=conv(num3,num4);
den5=conv(den3,den4);


TF5=tf(num5,den5);

rn=roots(num5);

rd=roots(den5);
%%%%%%%%after cancelling the similar terms of numenator and
denomator:
rnn=[11.6646;-11.9723;-0.0123];
dnn=[0;-2.1294 + 2.4415i;-2.1294 - 2.4415i;-0.0165 + 0.2607i;-0.0165 -
0.2607i];





num=poly(rnn);
den=poly(dnn);
%%%%num=[1.0000 0.3200 -139.6483 -1.7177];
```

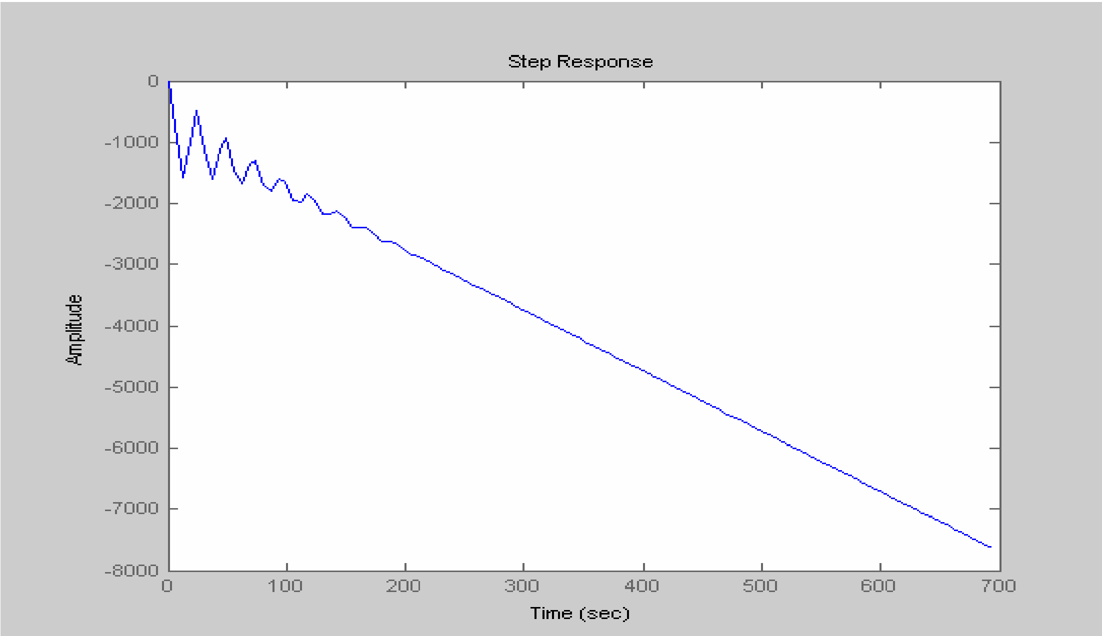
```
%%%%den=[1.0000 4.2918 10.7040 0.6370 0.7162 0];
num=[4.182 1.344 -584 -7.187];
den=[1 4.354 10.69 0.6385 0.7282 0];
TF=tf(num,den)
```

The output gives the transfer function for this system that controls altitude hold autopilot to be:

$$TF = \frac{4.182\ s^3 + 1.344\ s^2 - 584\ s - 7.187}{s^5 + 4.354\ s^4 + 10.69\ s^3 + 0.6385\ s^2 + 0.7282\ s}$$

➔ We look to the response of this system by using this Matlab command:

```
num=[4.182 1.344 -584 -7.187];
den=[1 4.354 10.69 0.6385 0.7282 0]
TF=tf(num,den)
>> step(TF)
```

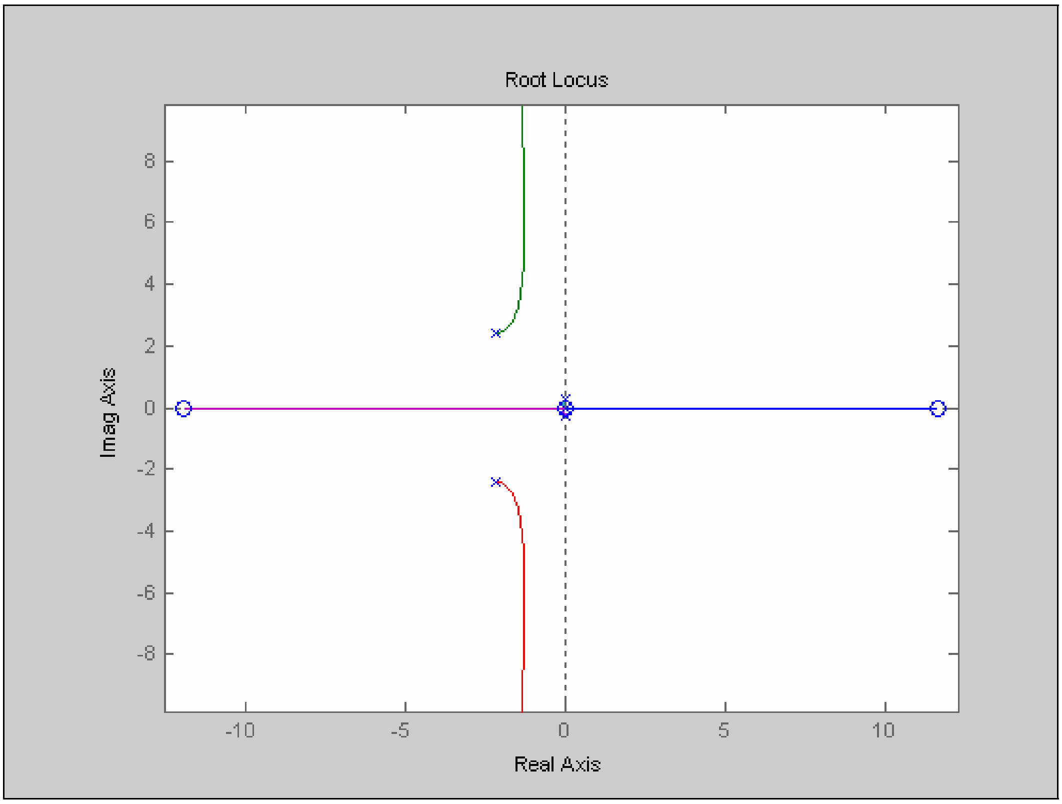


The figure shows that the system is not stable and we need to stabilize it by designing a stabilizer.

We find the root locus of the previous transfer function; we use the following Matlab command:

```
num=[4.182 1.344 -584 -7.187];
den=[1 4.354 10.69 0.6385 0.7282 0];

TF=tf(num,den)

rlocus(num,den)
```

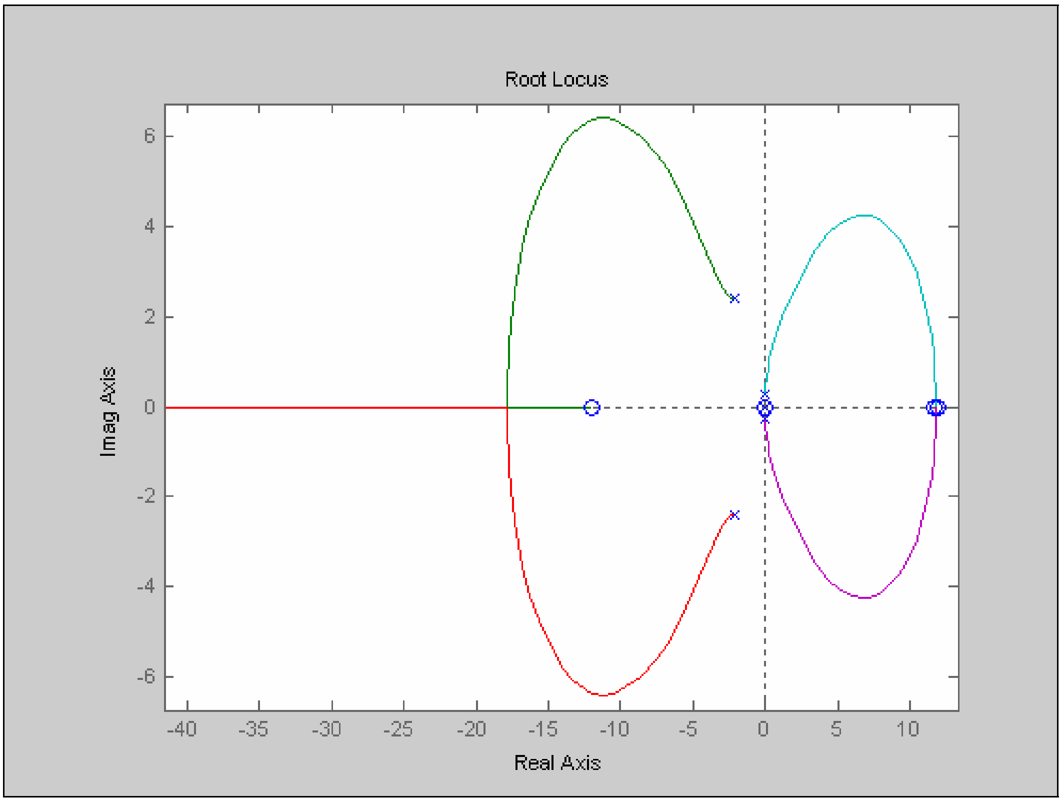
The root locus is:



By looking to the graph, the root loci are moving to the right half plane, we need to add a zero to pull it to the stable region. To do so, we the following Matlab command:

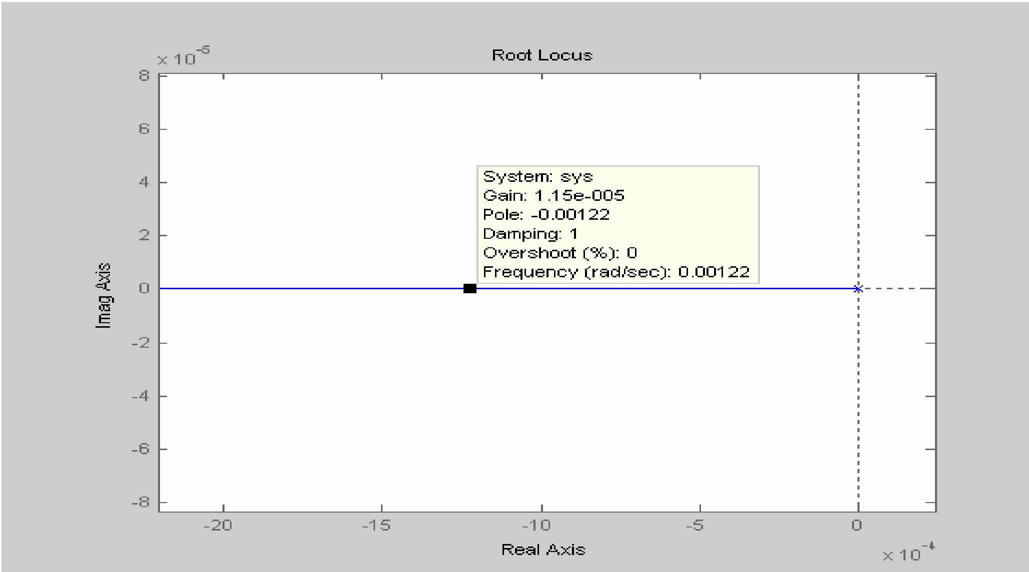
```
%%%%by adding a zero ti the right of our zero to bring the root locus to the stable part:
%%%%% THE ZERO IS (S-12)
Nc=[1 -12];
NGH=conv(Nc,num);
DGH=den;
rlocus(NGH,DGH);
rlocfind(NGH,DGH);%%%%to find the location of the gain K
```

The root locus after adding the zero is:



In this figure we find a value for the gain k to make the system stable. Of course we look for a gain which is small and that makes the system stable.

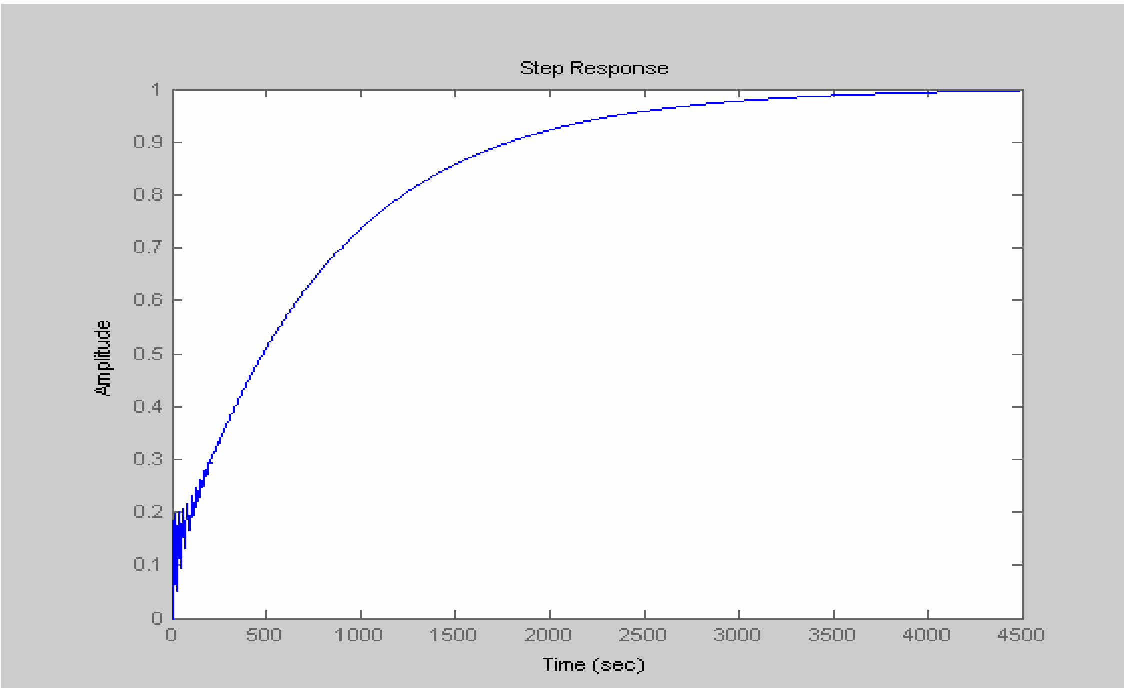
By zooming in:



As shown in the figure the value of gain k = 1.15e-5 (which is very small and that means very cheap).

We look to the response now after the gain is added and the system is having closed loop. By using this Matlab command:

```
[N,D]=cloop(1.15e-5*NGH,DGH,-1);
step(N,D);
```



By looking to the response, it shows that the system is stable.

So the classical control theory provided us a stable system after adding this gain.

Part 4: (Designing an altitude hold autopilot by using modern control theory).

We add the fourth state which represents the altitude (h) to have fifth order system as our transfer function showed.
The new system matrices are:

```
 A=[-0.03893 5.4535 -0.40758 -9.8 0
    -0.008376 -1.2854 0.9764 -0.000001089 0
    0.013905 -6.7369 -3.0292 0 0
    0 0 1 0 0
    0 -1 0 1 0];

B=[-0.60796
    -0.092937
    -10.0601
    0
    0];

C=[0 0 0 0 1]

D=[0]

R=rank([B,A*B,A*A*B,A*A*A*B,A*A*A*A*B]);
```

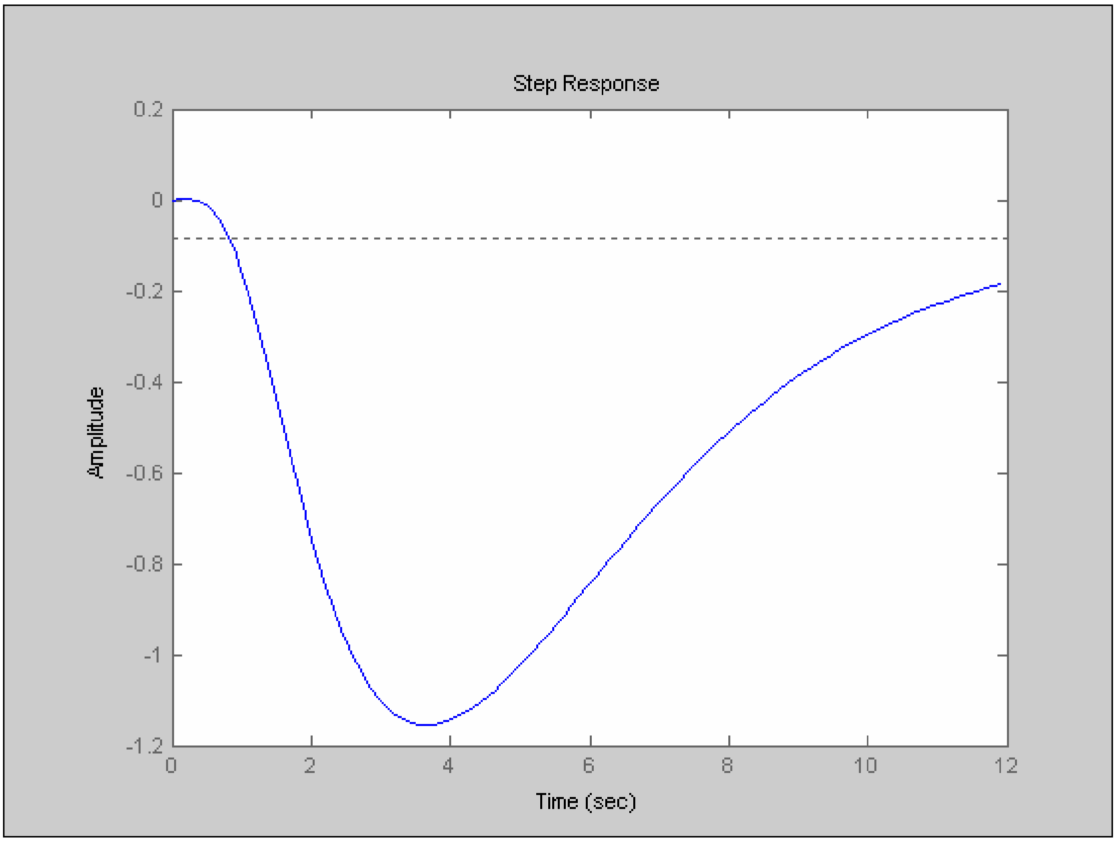
The output showed that the rank is 5, which is equal to order of the system. So all the four states are controlled.

Now we want to satisfy this aircraft flying qualities in this design. Sine this aircraft is considered to be (class I, category B), we choose the value of damping to be 0.6 and settling time to be 2.5 sec.
According to these values we place our new poles by using this Matlab command:

```
K=place(A,B,[-0.5+0.002i,-0.5-0.002i,-1.602+2.14i,-1.602-2.14i,-1])
E=eig(A-B*K)
```

Now we check the time response for altitude by using this Matlab command:

```
step(A-B*K,B,C,D);
```



As the figures shows, the altitude is stable.

Conclusion:

The purpose was to design an altitude hold autopilot for this aircraft by using two methods of design, the first was by the classical control theory and the second was by the modern control theory after adding the gains according to the flying qualities. The purpose was accomplished by the help of Matlab program and as a conclusion, we can say that using modern control theory we can control any system and easy to design but with classical control theory that could be difficult and sometimes impossible.

Appendix:

The whole design Matlab code for both classical and modern control theories is:

```
A=[-0.03893 5.4535 -0.40758 -9.8
   -0.008376 -1.2854 0.9764 -0.000001089
   0.013905 -6.7369 -3.0292 0
   0 0 1 0];

B=[-0.60796
   -0.092937
   -10.601
   0];

u0=45;
C2_alpha=[0 1 0 0];
C1_q=[0 0 1 0];
D=0;

[num1,den1]=ss2tf(A,B,C1_q,D);

[num2,den2]=ss2tf(A,B,C2_alpha,D);




T1_q=SS2TF(A,B,C1_q,D);num1=[0  -10.6010  -13.4216   -1.0426    0.0000];
                  den1=[1.0000    4.3535   10.6910    0.6385    0.7282];


T2_alpha=SS2TF(A,B,C2_alpha,D);num2ns=[0   -0.0929  -10.6309   -0.4435   -
0.8828];
                        den2ns=[1.0000    4.3535   10.6910    0.6385    0.7282];


 TF1=tf(num1,den1);
 TF2ns=tf(num2ns,den2ns);
 s=[1 0];
 num2=conv(s,num2ns);
 den2=den2ns;
 TF2=tf(num2,den2);
 TF3=TF1-TF2;
 num3=[0.0929 0.4343 -11.85 -56.28 -139.4 -9.973 -9.553 -0.1164 0];
 den3=[1 8.707 40.33 94.36 121.3 19.99 15.98 0.9299 0.5303];

 num4=[u0];den4=[1 0 0];
 TF4=tf(num4,den4);
```

```
num5=conv(num3,num4);
den5=conv(den3,den4);


TF5=tf(num5,den5);

rn=roots(num5);

rd=roots(den5);
%%%%%%%%after cancelling the similar terms of numenator and denomator:
rnn=[11.6646;-11.9723;-0.0123];
dnn=[0;-2.1294 + 2.4415i;-2.1294 - 2.4415i;-0.0165 + 0.2607i;-0.0165 - 0.2607i];




num=poly(rnn);
den=poly(dnn);


%%%%num=[1.0000 0.3200 -139.6483 -1.7177];
%%%%den=[1.0000 4.2918 10.7040 0.6370 0.7162 0];

num=[4.182 1.344 -584 -7.187];
den=[1 4.354 10.69 0.6385 0.7282 0];


TF=tf(num,den)



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%CLASSICAL CONTROLL
THEORY%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
rlocus(num,den)

%%%%by adding a zero ti the right of our zero to bring the root locus to the stable
part:
%%%%% THE ZERO IS (S-12)
Nc=[1 -12];
NGH=conv(Nc,num);
DGH=den;
rlocus(NGH,DGH);
rlocfind(NGH,DGH);%%%%to find the location of the gain K

%%%%%%%selected point was (p=-0.00122)
%%%%%%%%%%%value of gain    (K=1.15e-5)
%%%%%%%%%%%%frequency=(0.00122 rad/sec)
%%%%%%%%%%%%%% damping=1 (satisfyes class 1,category B)
```

```
%%%%%NOW WE CHECK AFTER ADDING THE VALUE OF GAIN FOR THE
SYSTEM:
[N,D]=cloop(1.15e-5*NGH,DGH,-1);
step(N,D);
%%%%%%%FROM THE GRAPH, ITI STABLE


%%%%%%%%%%%%%%%%%%%%%%%%%%%Modern control (Full state
feedback)%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%the nem system matrices as
follows%%%%%

A=[-0.03893 5.4535 -0.40758 -9.8 0
   -0.008376 -1.2854 0.9764 -0.000001089 0
   0.013905 -6.7369 -3.0292 0 0
   0 0 1 0 0
   0 -1 0 1 0];

B=[-0.60796
   -0.092937
   -10.0601
   0
   0];

C=[0 0 0 0 1]
D=[0]

R=rank([B,A*B,A*A*B,A*A*A*B,A*A*A*A*B]);
%%% by looking to the flying qualities for this aircatft we find that the damping = 0.6
and we prefer out settling to aaaaabout 2.5 second


K=place(A,B,[-0.5+0.002i,-0.5-0.002i,-1.602+2.14i,-1.602-2.14i,-1])
E=eig(A-B*K)
%%%%%to check the time response for altitude:

step(A-B*K,B,C,D);
```